Temporal Process Discovery in Many Guises

Naren Ramakrishnan, Debprakash Patnaik, and Vandana Sreedharan, *Virginia Tech*

The dynamics of important temporal processes can now be automatically reconstructed from data.

Timing is everything. Neurons fire in time-locked fashion to propagate signals and form memories. The cell division cycle, the process by which an adult cell divides into two daughter cells, is carefully orchestrated by rises and decays of regulating protein concentrations.

Engineered systems, like data centers, also involve temporal coordination. A few months back, the music service Last.fm had to shut down temporarily due to overheating in its data center. Dynamically starting chillers, shutting them down, and commissioning new ones is critical to keeping such services running continually.

THREE COMMUNITIES

At least three AI and AI-related communities are interested in time and temporal modeling.

One is the machine learning and data mining community, which extracts temporal patterns and relationships from data streams. Available data mining techniques infer many types of patterns such as frequent episodes ("Event A seems to occur frequently and is typically followed by B five milliseconds later, and then by C three milliseconds later") and probabilistic networks ("It looks like either A or B needs to happen before C can happen").

A second community casts processes in a suitable representation and then reasons deductively about the dynamics of events. James Allen's interval taxonomy (J.F. Allen, "Maintaining Knowledge about Temporal Intervals," *Comm. ACM*, Nov. 1983, pp. 832-843) is a famous example of temporal representation and reasoning. It models 13 types of relations between intervals to aid in planning tasks. We can model Amtrak's rail system with these relations and use reasoning algorithms to provably find a way to get to Boston from Atlanta.

Finally, the model-checking community uses a body of algorithms to verify reactive systems—systems that interact with their environment over time (E.M. Clarke Jr., O. Grumberg, and D.A. Peled, *Model Checking*, MIT Press, 1999).

Given a state machine description of the system—for example, a washing machine—and a property described in temporal logic ("foam must not overflow when washing is initiated"), model checking can be used to verify preservation of the property. Even better, if the property could be violated, model checking will highlight a suitable counterexample: "To get the foam to overflow, load the machine this way, add this much detergent, and turn the knob to Wool."

The distinction between deductive reasoning and model checking is subtle but important. In the former, we apply "first principles" inference rules systematically from a starting set of axioms. While this can accommodate infinite state spaces, deductive reasoning involves the manual trouble of modeling the system. This is undoubtedly overkill if we are interested in only verifying specific properties. Model checking is a brute-force approach that focuses on finite state spaces but is also more automated.

Today there are approaches that mix deduction with model checking. However, both schemes require conceptualizing the temporal dynamics, either as a state-transition diagram or a complete axiomatization. This is where data mining comes in. If we can use data mining to extract conceptual models of the dynamics, we

0018-9162/09/$26.00 © 2009 IEEE

can immediately begin posing interesting temporal questions of these models.

Some questions in particular are very compelling. In a complex system, how do entities group and dynamically regroup over time? What process relationships does the data reflect? How can we "steer" the system toward interesting or desirable states?

ONE COMMON GOAL

These questions have immediate applications in neuroscience, systems biology, and data center management. "Entities" can respectively refer to areas of neurophysiological activity; genes, proteins, and metabolites; and thermal sensors. "Processes" can denote information-coding pathways in a neural system, biological regulation in a cell, or cooling propagation in a data center. Hence, a unified approach to process modeling from data can help make important strides in all of these domains.

A major catalyst for such research is the emergence of new technologies to measure and monitor temporal processes, as Figure 1 shows.

Computational neuroscience is undergoing a data revolution similar to what biology first began to experience in the early 1990s. Scientists can now record simultaneous spike trains from neuronal tissue using multielectrode arrays.

In biology, interest in temporal modeling used to be largely driven by data. Now, however, researchers seek to understand biology at a system level by constructing and simulating models of key biological processes—for example, the cell division cycle and even entire signaling pathways. These models thus also serve as the source of data streams.

Finally, data centers, notwithstanding the scale of the information they serve, have themselves become producers of data. A data center can be viewed as a massive sensor network because it tracks numerous physical variables such as workload, utilization, temperature, humidity, airflow, and power.

The goal of temporal process discovery is threefold:

- redescribe event streams,
- integrate formal logic with data mining, and
- control dynamic processes.

In essence, the idea is to infer entire temporal models from data and reason with these models with an eye toward comprehension or control. While the nature of input data can vary between symbolic and continuous-valued, the objective is often the same: to raise the abstraction to capture higher-level temporal concepts.

One way to realize temporal redescriptions is through a cluster dynamics approach.

TEMPORAL REDESCRIPTION

Redescription refers to the idea of restating some given information in a different vocabulary, often to yield insight. The term was coined for general learning and mining contexts (L. Parida and N. Ramakrishnan, "Redescription Mining: Structure Theory and Algorithms," *Proc. 20th Conf. Artificial Intelligence*, AAAI, 2005, pp. 837-844), but here we use it specifically in the context of temporal data.

Redescribing a temporal event stream elevates the vocabulary to specify which events occur before which others, identify the "checkpoints" that must be satisfied (and when), and determine whether there can be alternative pathways of time-series progression.

One way to realize temporal redescriptions is through a *cluster dynamics* approach that views the system entities as forming groups ("clusters") that are dynamically revised at important time-series stages. To identify such groups automatically from data, we can cluster the entities or the time points themselves.

The first approach, illustrated by Figure 2, initially strips temporal information out of the data and clusters the entities to identify dense regions of state space. The original time-series data is redescribed in terms of these clusters, thereby restoring the temporal information. By using clusters of vectors to define the states and transitions between these states to determine the system trajectories, we can reconstruct key dynamic features such as linear-state progressions and even higher-level features such as oscillations.

Alternatively, temporal redescription can be viewed as a task of segmenting the time-series data. Each segment is modeled as a mixture of clusters such that segment boundaries involve significant regrouping and redefinition of clusters (S. Tadepalli et al., "Simultaneously Segmenting Multiple Gene Expression Time Courses by Analyzing Cluster Dynamics," *J. Bioinformatics and Computational Biology*, Apr. 2009, pp. 339-356).

These forms of redescriptions essentially "symbolize" continuous-valued data into a form suitable for regular temporal data mining algorithms. In particular, a frequent-episode mining algorithm can be applied to extract high-level motifs underlying the data (D. Patnaik et al., "Sustainable Operation and Management of Data Center Chillers Using Temporal Data Mining," *Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, ACM Press, 2009, pp. 1305-1314).

FROM PATTERNS TO MODELS

Extracted temporal patterns become the raw material for temporal model building.

Gantt charts, for example, capture temporal compartmentalization of processes. They break up the time series into segments and describe what happens in each segment so that approaches like Allen's taxonomy can then be applied.

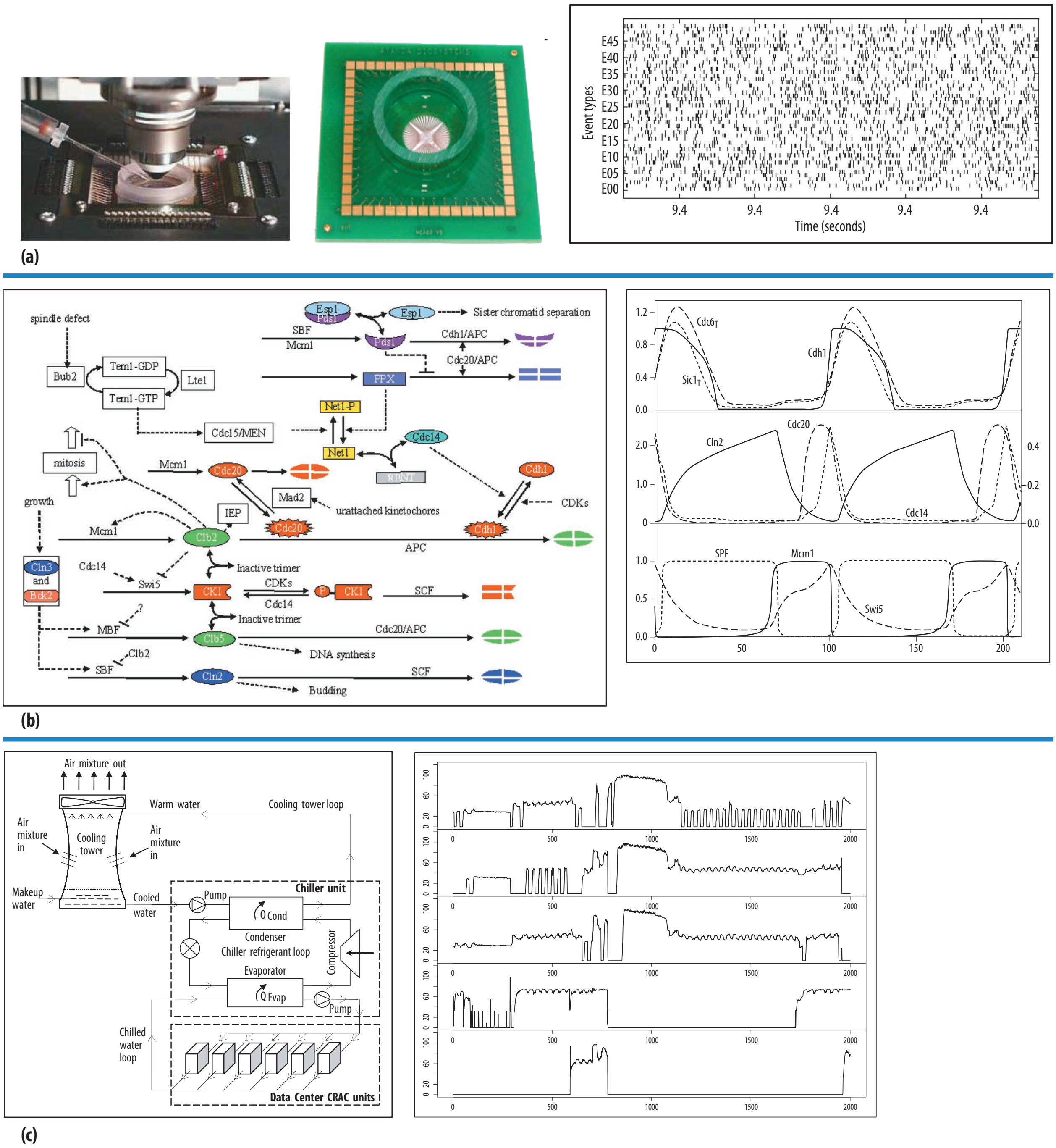

Figure 1. Temporal data sources. (a) A multielectrode array records spike trains from neuronal tissue. (b) Systems biology models simulate time-course trajectories of protein levels. (c) Sensor networks track utilization of chillers in a data center. Courtesy of J.J. Tyson, M. Marwah, and R. Sharma.

A Kripke diagram is basically a state-transition diagram with some bells and whistles. States are labeled with propositions that hold true in that state so that domain-specific questions can be posed in terms of temporal logic operators over the paths in the diagram.

Thus, formal temporal logic helps endow data mining algorithms with increased expressiveness about temporal modeling.

Using a suitable vocabulary of temporal logic operators, such as those used in computation tree logic, we can make inferences about the satisfaction or refutability of temporal logic formulae (N. Ramakrishnan, M. Antoniotti,

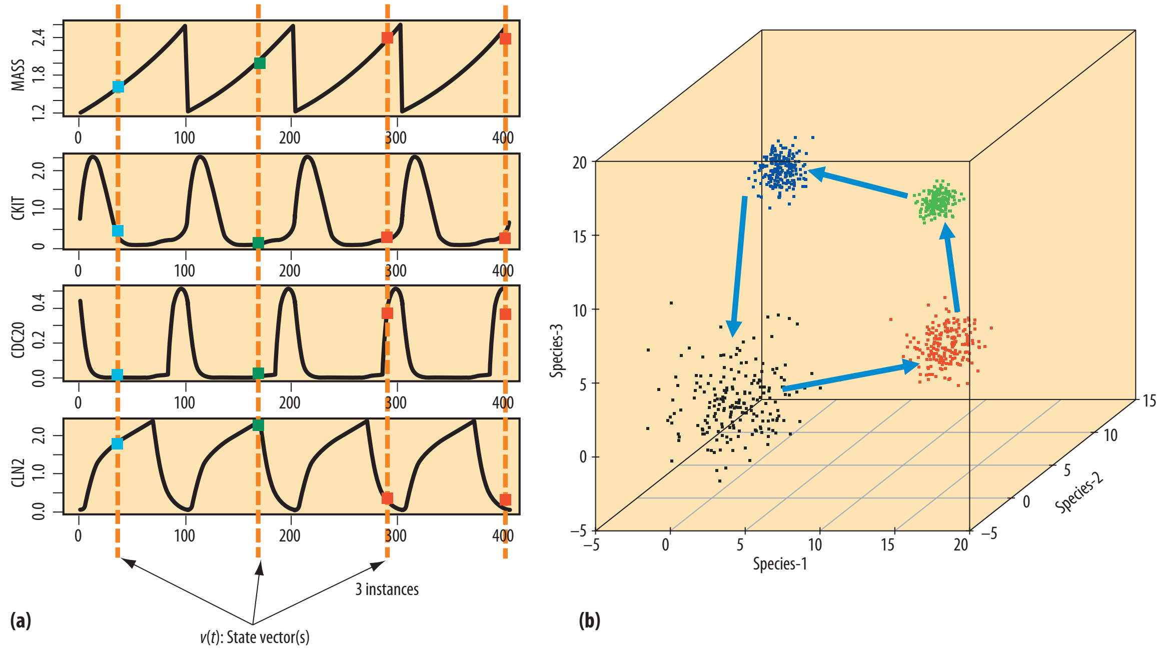

Figure 2. Redescribing a multivariate time course dataset by (a) stripping out temporal information and clustering the data points, and (b) restoring the temporal information. This example is from systems biology.

and B. Mishra, "Reconstructing Formal Temporal Models of Cellular Events Using the GO Process Ontology," *Proc. 8th Ann. Bio-Ontologies Meeting,* 2005; http://people.cs.vt.edu/~naren/papers/GOALIEbioontologies.pdf).

Figure 3 illustrates a complete temporal model inferred from data. The model integrates data from the budding yeast cell cycle (the "normal" or wild-type condition) with data obtained by knocking out the cell cycle in different ways (the "mutants"). It is known biologically that mutants inactivate important pathway components for progression through the cell cycle and hence arrest cells in some state without completing the cycle. The goal is to see if the model reflects this a priori domain knowledge.

In Figure 3, the mutants are denoted by experimental conditions 1-4 and the normal condition by experiment 5. To identify the states, we stripped out temporal information from all datasets and restored them while analyzing cluster dynamics. The middle of the figure depicts these states and labels transitions between them with the experiments under which they were observed.

It is clear that there is a central cell cycle pathway and that each mutant forks off a separate, irreversible, transition to abnormal states. Note how each such fork is labeled with a unique, mutant condition.

It is a straightforward step to convert Figure 3 into a Kripke model by labeling each node in the diagram with properties and processes known to be true in that state. Ontologies such as GO (Gene Ontology) help in such labeling by providing a controlled vocabulary for describing active sets of genes and proteins.

GAINING CONTROL

Ultimately, the goal of temporal process modeling is to understand the modeled system's transfer function to proactively steer the system toward desirable states.

Objective criteria for such control can be defined in many ways—for example, to stimulate a given pathway (neuroscience); circumvent, halt at, or disrupt a given process (systems biology); or guide the system's dynamics toward regions of increased energy efficiency (data centers). In all such cases, the goal of dynamic control can be captured in terms of the underlying system entities that must be perturbed, and how.

The idea of control is especially attractive when we consider the interplay among multiple temporal processes.

For instance, it is well known that the yeast cell division cycle (YCC) and the yeast metabolic cycle (YMC) are interleaved, but the precise mechanisms are not well characterized. Constructing a model such as that shown in Figure 3 using both YCC and YMC datasets will not only reveal the temporal coordination between them but will also help clarify whether temporal "hardwiring" is manifest in these processes.

Can we make the system adopt an aberrant cell state—for example, suspended animation—or make it proceed along an artificial, chosen, event

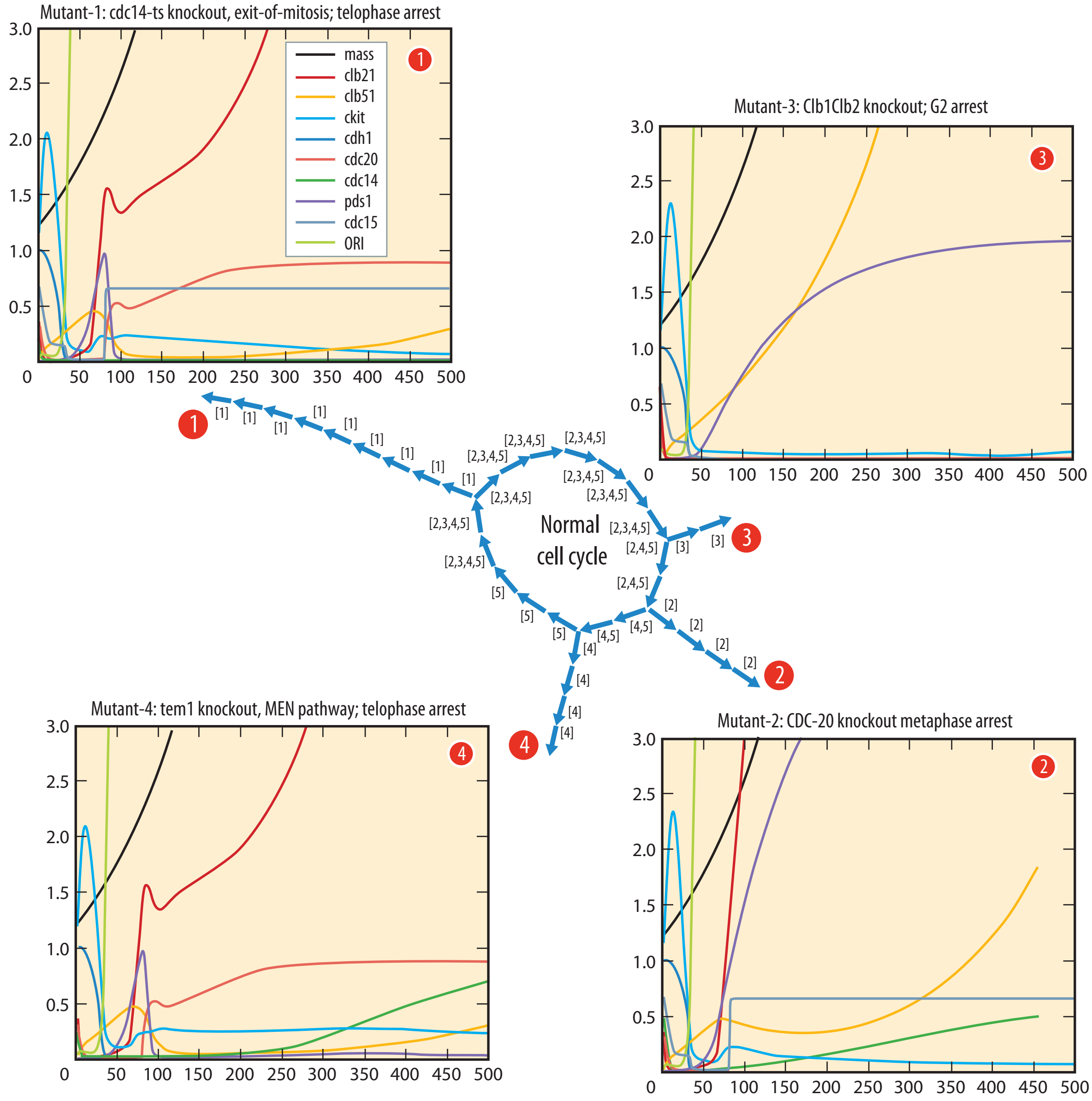

Figure 3. State progression of normal versus mutant yeast cell cycle models, inferred automatically from data. Each mutant forks off a separate, irreversible transition to abnormal states.

sequence? We are initiating studies to investigate such questions.

Multiple viewpoints of temporal modeling have clearly begun to coalesce. The meeting points often involve some marriage of data-driven and model-driven reasoning. Hot application areas such as robotics and computational biology are contributing to these marriages.

Nevertheless, the field is still nascent. Researchers who can bridge the inductive emphasis of data mining and machine learning with the deductive/verification viewpoint of logic reasoning and model checking can readily contribute to the growing excitement. ■

Naren Ramakrishnan *is a professor of computer science at Virginia Tech. Contact him at naren@cs.vt.edu.*

Debprakash Patnaik *is a PhD student in computer science at Virginia Tech. Contact him at patnaik@vt.edu.*

Vandana Sreedharan *is a PhD student in the Genetics, Bioinformatics, and Computational Biology program at Virginia Tech. Contact her at vsree007@vt.edu.*